

## AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A system comprising:  
a non-volatile data storage device, ~~configure~~ configured as one or more storage regions, to store one or more bytes of data;  
a program store communicatively coupled to the non-volatile data storage device, the program store to store one or more processor-readable instructions to ascertain the validity of data stored in the non-volatile storage device and if invalid to replace the data with an earlier stored valid image of the data; and  
a processing unit coupled to the non-volatile data storage device and program store, to read and process the one or more instructions in the ~~process-program~~ store.
2. (Previously Presented) The system of claim 1 wherein the processing unit is configured to process the instructions in the program store as part of a start-up procedure.
3. (Currently Amended) The system of claim 1 wherein the data stored in the non-volatile data ~~store~~ storage device is a Basic Input Output System (BIOS) ~~for a processing device~~.
4. (Original) The system of claim 1 wherein the processor-readable instructions in the program store ascertain the validity of the data stored in the non-volatile storage device on a region by region basis.
5. (Original) The system of claim 1 wherein the earlier stored valid image of the data is stored in a location that cannot be modified without system authorization.
6. (Original) The system of claim 5 wherein system authorization includes employing a system interface to perform modifications to the data stored in the non-volatile data storage device.
7. (Original) The system of claim 1 wherein ascertaining the validity of the data stored in the non-volatile storage device includes

determining if the current data in the non-volatile storage device is different than the earlier stored valid image of the data.

8. (Original) The system of claim 1 wherein ascertaining the validity of the data stored in the non-volatile storage device includes

determining if an integrity metric corresponding to the current data in the non-volatile storage device is different than the same integrity metric corresponding to the earlier stored valid image of the data.

9. (Original) The system of claim 1 further comprising:

generating a copy the current data in the non-volatile storage device if an authorized application modifies the current data; and  
storing the copy as a valid image of the current data.

10. (Previously Presented) A method comprising:

reading current content stored in a non-volatile storage device;  
determining if the current content has been modified without authorization; and  
replacing the current content with a previously stored valid image of the content if the current content is determined to have been modified without authorization.

11. (Previously Presented) The method of claim 10 further comprising:

reading the valid image of the previously stored content; and  
comparing the previously stored content to the current content to determine if the current content has been modified.

12. (Original) The method of claim 10 wherein determining if the current content has been modified without authorization includes

comparing a previously stored checksum, corresponding to the valid image of the previously stored content, and the checksum corresponding to the current content.

13. (Original) The method of claim 10 wherein determining if the current content has been modified without authorization includes

comparing a previously stored cyclic redundancy check value, corresponding to the valid image of the previously stored content, and the cyclic redundancy check value corresponding to the current content.

14. (Currently Amended) The method of claim 10 wherein determining if the current content has been modified without authorization includes  
comparing a previously stored bit mask, corresponding to the valid image of previously stored content, and the bit mask corresponding ~~bits of~~ the current content.
15. (Original) The method of claim 10 further comprising:  
storing a valid image of the current content for later use.
16. (Original) The method of claim 10 wherein the content is read from the non-volatile storage device as part of a start-up procedure.
17. (Previously Presented) A method comprising:  
arranging a non-volatile storage device into one or more storage regions;  
generating an integrity metric corresponding to valid content stored in a first region of the non-volatile storage device; and  
storing the integrity metric to later determine if the content in the first region has been modified without authorization.
18. (Original) The method of claim 17 further comprising:  
comparing a previously stored integrity metric, corresponding to an earlier version of the content stored in the first region, to a newly calculated integrity metric corresponding to the current content stored in the first region to determine if an unauthorized modification has occurred.
19. (Original) The method of claim 17 further comprising:  
replacing the first region with an earlier version of the content therein if it is determined that there was an unauthorized modification.
20. (Previously Presented) A method comprising:  
arranging a non-volatile storage device into one or more storage regions; and  
comparing current content in a first region to an earlier stored image of the content in the first region; and  
replacing the current content stored in the first region with the previously stored content of the first region if it is determined that there was an unauthorized modification of the current content.

21. (Original) The method of claim 20 wherein the method is implemented as part of a start-up procedure.
22. (Currently Amended) The method of claim 20 wherein the non-volatile storage device is arranged into one or more logical regions, each region ~~of~~ having one or more bytes.
23. (Currently Amended) A method comprising:  
arranging a non-volatile storage device into one or more storage regions;  
verifying that the content in the non-volatile storage device is valid; and  
encrypting the content in a first region by use of a first encryption key to protect it from unauthorized access.
24. Canceled.
25. (Currently Amended) The method of claim 23 further comprising:  
protecting the content of ~~the content of~~ a second region with a second encryption key.
26. (Previously Presented) A machine-readable medium having one or more instructions for protecting content in a non-volatile storage device against unauthorized use, which when executed by a processor, causes the processor to perform operations comprising:  
reading current content stored in a non-volatile storage device;  
determining if the current content has been modified without authorization; and  
replacing the current content with a previously stored image of the content if the current content is determined to have been modified without authorization.
27. (Original) The machine-readable medium of claim 26 wherein determining if the current content has been modified without authorization includes  
reading an image of previously stored content; and  
comparing the previously stored content to the current content to determine if the current content has been modified.

28. (Original) The machine-readable medium of claim 26 wherein determining if the current content has been modified without authorization includes

comparing a previously stored checksum corresponding to a valid image of previously stored content and the checksum corresponding to the current content.

29. (Original) The machine-readable medium of claim 26 wherein determining if the current content has been modified without authorization includes

comparing a previously stored cyclic redundancy check value corresponding to a valid image of previously stored content and the cyclic redundancy check value corresponding to the current content.

30. (Currently Amended) The machine-readable medium of claim 26 wherein determining if the current content has been modified without authorization includes

comparing a previously stored bit mask corresponding to a valid image of previously stored content and the bit mask corresponding ~~bits of~~ to the current content.